Attorney Docket No. IBM/244
Confirmation No. 9970

PATENT

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

---

***Ex parte*** Timothy Roy Block, Bob Richard Cernohous and John Christian Unterholzner

---

Appeal No. _____
Application No. 10/626,086

---

# APPEAL BRIEF

---

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | | | |
|---|---|---|---|
| Applicant: | Timothy Roy Block et al. | Art Unit: | 2155 |
| Application No.: | 10/626,086 | Examiner: | Thuong Nguyen |
| Filed: | July 24, 2003 | | |
| For: | CLUSTER DATA PORT SERVICES FOR CLUSTERED COMPUTER SYSTEM | | |

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

# APPEAL BRIEF

## I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation, of Armonk, New York.

## II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

## III. STATUS OF CLAIMS

Claims 1, 3-16 and 18-36 are pending in the Application, stand rejected, and are now on appeal. Claims 2 and 17 have been canceled.

## IV. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection mailed May 22, 2006.

## V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant's invention is generally directed to an apparatus, program product and method that utilize cluster data port services within a cluster infrastructure to provide reliable and

efficient communications between nodes in a clustered computer system. (Application, page 4, lines 2-5). In particular, cluster data port services consistent with the invention present an abstracted transport service that encapsulates and manages the establishment of multiple logical connections between a source node, a target node and one or more backup nodes in such a manner that a cluster data port is effectively utilized as single data port from the perspective of a user program. (Application, page 4, lines 5-9).

Specifically, the claimed subject matter recited in claim 1 is directed to a method of communicating between a source and a target node in a clustered computer system (Application, page 6, lines 7-26, Fig. 1, elements 10, 12, 14 and 16). The claimed method includes establishing a cluster data port between the source node and a target node, where the cluster data port is configured to select among a plurality of connection paths between the source node and the target node (Application, page 7, lines 3-12, Fig. 1, elements 22 and 24, *see also*, Application, page 19, line 19 to page 20, line 27 and Fig. 8), and to selectively switch over data flow from the target node to a backup target node (Application, page 7, lines 12-18, Fig. 1, elements 14, 16, *see also*, Application, page 23, lines 14-25 and Fig. 11). The claimed method also includes communicating data from the source node to the target node using the cluster data port (Application, page 21, line 12 to page 22, line 15 and Fig. 9).

In addition, claim 1 recites that establishing the cluster data port includes establishing multiple concurrent logical connections between the source node and the target node (Application, page 16, line 13 to page 17, line 2, Fig. 6, blocks 120-136), where each logical connection is configured to communicate data over a connection path among the plurality of connection paths (Application, page 7, lines 2-7, Fig. 1, blocks 20, 22).

With respect to independent claim 16, the apparatus includes a memory and at least one processor (Application, page 7, lines 28-29, Fig. 2, blocks 26, 28) and program code (Application, page 7, line 8 to page 10, line 1, Fig. 3, blocks 40-60) configured to perform the steps described above in connection with claim 1. Likewise, with respect to independent claim 31, the clustered computer system includes a plurality of nodes (Application, page 6, lines 7-10, Fig. 1, blocks 10-16) and a cluster data port resident on at least one of the plurality of nodes and configured in the manner described above in connection with claim 1. With respect to

independent claim 35, the program product includes program code (Application, page 7, line 8 to page 10, line 1, Fig. 3, blocks 40-60) configured to perform the steps described above in connection with claim 1 and a signal bearing medium (Application, page 10, lines 1-5.[1]

A cluster data port as set forth in the claims supports both the ability to select among a plurality of connection paths between a source node and a target node, and the ability to selectively switch over data flow from a target node to a backup target node (Application, page 4, lines 10-13). As such, the failure of a given connection path, or alternatively, the failure of a given target node, may be rectified by such a cluster data port, often in a manner that is effectively hidden from any user programs that utilize the service, or in the least, in a manner that such user programs are freed from having to be involved in managing the underlying communication layer protocols (Application, page 4, lines 13-17). In addition, load balancing may also be supported by a cluster data port when multiple connection paths, and thus multiple logical connections, exist between two nodes (Application, page 4, lines 17-19).

## VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1, 16, 31 and 35 stand rejected under 35 U.S.C. § 112, first paragraph.

B. Claims 1, 16, 31 and 35 stand rejected under 35 U.S.C. § 112, second paragraph.

C. Claims 1-4, 6, 8-10, 12, 14-19, 21, 23-25, 27 and 29-36 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2002/0161923 to Foster (*Foster*).

D. Claims 5, 7, 11, 13, 20, 22, 26 and 28 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Foster* in view of U.S. Patent No. 6,587,866 to Modi (*Modi*).

---

[1]Applicants submit that the foregoing provides ample support for the claimed subject matter under 37 CFR §41.37(c)(1)(v). Moreover, as none of the claims recite any means plus function or step plus function elements, Applicants submit that no identification of such elements is required pursuant to 37 CFR §41.37(c)(1)(v).

## VII. ARGUMENT

Applicants respectfully submit that the Examiner's rejections of claims 1, 3-16 and 18-36 are not supported on the record, and should be reversed. Applicants' remarks in rebuttal to the Examiner's rejections are presented below. Applicants will hereinafter address the Examiner's rejections in the order presented in the Final Office Action. Within the discussion of each rejection, the various claims that are the subject of the Examiner's rejections will further be addressed in order, starting with the independent claims, and then addressing various dependent claims reciting additional subject matter that is distinguishable from the prior art of record. In some cases, specific discussions of particular claims are not made in the interests of streamlining the appeal. The omission of a discussion with respect to any particular claim, however, should not be interpreted as an acquiescence as to the merits of the Examiner's rejection of the claim, particularly with respect to claims reciting features that are addressed in connection with the rejections applied to other claims pending in the appeal.

### A. Claims 1, 16, 31 and 35 were improperly rejected under 35 U.S.C. § 112, first paragraph.

Claims 1, 16, 31 and 35 are rejected as failing to comply with the written description requirement, asserting that the claims contain subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention. In particular, the Examiner asserts that the specification does not disclose the claimed feature of establishing multiple concurrent logical connections between a source node and a target node.

As shown, however, in Fig. 1, the specification discloses an embodiment where each node 12, 14, 16 includes multiple network connections 20 that communicate over multiple connection paths 22. For example, between source node 12 and primary target node 14, multiple connection paths 22 exist between the two nodes, with multiple network connections 20 defined within each node 12, 14 to communicate over the multiple connection paths. As discussed at page 7, lines 2-5 of the Application, "by providing multiple connection paths 22 from a node 12, 14, 16 to network 18, multiple logical connections may be made between any pair of nodes."

(*emphasis added*). As such, the specification explicitly states that multiple logical connections may be established between a source node and a target node.

With respect to the fact that these logical connections are "concurrent" in nature, while the term "concurrent" is not expressly used in the specification, the specification as a whole would be interpreted by one of ordinary skill in the art as teaching that the multiple logical connections established between two nodes are concurrent logical connections.

One of the principal features that are supported by some embodiments of the invention is load balancing of data communication over multiple logical connections between two nodes. The specification describes at page 7, lines 5-8 that the provision of multiple logical connections enables "both connection redundancy and failover (to recover from a failed connection), and optionally load balancing (to provide higher overall throughput), to be provided for each node in the clustered computer system." In addition, the specification notes at page 11, lines 6-11 that:

> [t]he cluster data port services . . . provide an abstracted transport service that encapsulates and manages the establishment of multiple logical or TCP connections to a designated target node with designated backup target nodes, which appears to the user (i.e., an operating system or kernel component) like a single data port or data pipe. Throughput is enhanced by data port services that enable workload balancing of traffic across multiple logical or TCP connections to a target node.

As one of ordinary skill in the art would readily appreciate, workload (or load) balancing across multiple logical connections as a practical matter necessitates that the logical connections be concurrent. "Concurrent" is defined in the American Heritage dictionary as "happening at the same time as something else."[2] Load balancing over multiple logical connections between two nodes involves the communication of data over the logical connections in parallel (i.e., concurrently) to improve the overall data throughput between the two nodes. To enable parallel data communication over multiple logical connections, therefore, the multiple logical connections must be concurrent (i.e., existing at the same time) in nature.

---

[2]The American Heritage Dictionary of the English Language, Fourth Edition (2004).

It is apparently the Examiner's position that while the specification discloses multiple logical connections between a source node and multiple target nodes, the specification only discloses that one logical connection be established between any two nodes at one time. However, were logical connections required to be serially opened and closed one such that only one logical connection between two nodes was open at a time, there would be no benefit to load balancing, since it would be impossible to communicate any data over the multiple connections in parallel. Indeed, due to the overhead associated with opening and closing the logical connections, the overall data throughput that would occur if data was communicated alternately on different connections would be lower than if one logical connection was used to communicate all of the data.

The Examiner takes the position that page 6, lines 2-6 and page 6, lines 21-23 of the specification and Fig. 6 only disclose single connections between individual pairs of nodes. However, Applicants submit that these passages do not support the Examiner's position. Of note, there is nothing in Fig. 6, nor in any of the accompanying text, that requires one logical connection between two nodes to be closed before another logical connection can be opened. However, the description of Fig. 6, in particular at page 16, lines 21-23, states that "[l]ogical or TCP connections are then established in block 132 by communicating one or more send and receive sockets from the source node to the target node" (*emphasis added*). In addition, Fig. 8, block 182, and the accompanying text at page 19, line 25 to page 2 describes load balancing performed "if multiple such connections paths have been established between the source and target nodes." It should also be noted that nowhere in the specification is there any discussion of closing one logical connection between two nodes prior to opening another logical connection between those two nodes, which would be necessary in order to operate in the manner suggested by the Examiner.

Applicants therefore submit that the specification provides ample support for the concept of concurrent logical connections, and that one of ordinary skill in the art would readily appreciate this concept. As such, the specification is enabling for this claimed feature, and the §112, first paragraph rejection is in error. Reversal of the Examiner's §112, first paragraph rejection is therefore respectfully requested.

## B. Claims 1, 16, 31 and 35 were improperly rejected under 35 U.S.C. § 112, second paragraph.

Claims 1, 16, 31 and 35 are also rejected as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicants regards as the invention. The Examiner is apparently unclear as to how to establish the multiple concurrent logical connections between the source node and the target node.

As discussed above in connection with the §112, first paragraph rejection, however, the concept of concurrent logical connections is well supported in the specification. Applicants therefore respectfully submit that the passages cited above in Section VII.A are not indefinite, and that the claims are fully compliant with §112, second paragraph. Reversal of the Examiner's §112, second paragraph rejection is therefore respectfully requested.

## C. Claims 1-4, 6, 8-10, 12, 14-19, 21, 23-25, 27 and 29-36 are not anticipated by *Foster*.

Claims 1-4, 6, 8-10, 12, 14-19, 21, 23-25, 27 and 29-36 are rejected as being anticipated by U.S. Patent Application Publication No. 2002/0161923 to *Foster*. Anticipation of a claim under 35 U.S.C. §102 requires that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros., Inc. v. Union Oil Co., 2 USPQ2d 1051, 1053 (Fed. Cir. 1987), *quoted in* In re Robertson, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999). Absent express description, anticipation under inherency requires extrinsic evidence that makes it clear that "the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill." Continental Can Co. v. Monsanto Co., 20 USPQ2d 1746, 1749 (Fed. Cir. 1991), *quoted in* In re Robertson at 1951. "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." Continental Can at 1749, *quoted in* In re Robertson at 1951.

Applicants respectfully submit that *Foster* does not disclose the various features recited in claims 1-4, 6, 8-10, 12, 14-19, 21, 23-25, 27 and 29-36, and as such, the rejections thereof should be reversed.

*Independent Claims 1, 16, 31 and 35*

Claim 1 recites a method of communicating between a source and a target node in a clustered computer system. The method includes establishing a cluster data port between the source node and a target node, and communicating data from the source node to the target node using the cluster data port. The cluster data port as claimed is configured to select among a plurality of connection paths between the source node and the target node, and to selectively switch over data flow from the target node to a backup target node. In addition, claim 1 recites that establishing the cluster data port includes establishing multiple concurrent logical connections between the source node and the target node, where each logical connection is configured to communicate data over a connection path among the plurality of connection paths.

Claims 16, 31 and 35 respectively recite an apparatus, a clustered computer system and a program product, each incorporating a cluster data port configured in a similar manner to that recited in claim 1.

As such, claims 1, 16, 31 and 35 are directed, in part, to a cluster data port that includes multiple concurrent logical connections between a source node and a target node.

In rejecting claim 1, the Examiner relies on *Foster*, and specifically paragraphs [0015], [0024], [0026], [0033] and [0036] and Fig. 1. However, the cited passages merely disclose a network-oriented routing system that dynamically controls routers in a switched network to communicate information between multiple nodes connected to the switched network. *Foster* establishes "virtual addresses", and controls mapping tables in various routing devices in a network such that communications associated with particular virtual addresses and received at a source port of a routing device are output to appropriate destination ports in the routing device that are associated with the specified virtual addresses.

*Foster*, however, establishes only one path at a time for a particular virtual address used by a source node. Paragraph [0015], for example, merely discloses managing an interconnect fabric of routing devices that are configured to establish a path that is uniquely associated with a virtual address. Paragraph [0024], on the other hand, discusses the possibility of identifying and configuring new paths, and assigning new paths to virtual addresses. However, it is evident from

*Foster* that in general only a single path is associated with a given virtual address used by a source node at any particular point in time.

To this extent that the Examiner may consider a virtual address to correspond to a logical connection, Applicants respectfully submit that *Foster* fails to disclose the establishment of multiple logical connections, much less multiple concurrent logical connections, between a specific pair of source and target nodes, as is required by claim 1. Indeed, paragraph [0017], which the Examiner relied upon in the first Office Action to reject the concept of multiple logical connections as originally recited in claim 2, discloses a network manager that can "re-identify" a path in response to a change in network topology. However, this disclosure is not directed to logical connections much less concurrent logical connections, as required by claim 1.

*Foster* does disclose, at paragraph [0022], the concept that a virtual address can be used by two different source nodes to communicate with the same destination node, and that the virtual address can therefore represent different paths that share a common sub-path. This concept is also referred to in paragraph [0024] in connection with the discussion of a virtual address being used to identify different paths. It is important to note, however, that from the perspective of any particular source/destination node pair, a virtual address in *Foster* is associated at any given time with only a single path. Since the virtual address is used by each routing device to route communications to a particular destination node, it can be seen how two source nodes, coupled to different source ports on the same routing device, could use the same virtual address to access the same destination node since the mapping table on the routing device would route the communications from each node to the same destination port, even though the communications may be received from different source ports on the device. By doing so, however, only one path would be established between the each source node and the common destination node. The fact that a virtual address may be used by two different source nodes to communicate with the same destination node over disparate paths (which still necessarily must share a sub-path) is therefore insufficient to anticipate the concept of multiple concurrent logical connections as recited in claim 1.

The Examiner specifically relies on paragraphs [0026], [0033] and [0036] and Fig. 1 for allegedly disclosing that the multiple logical connections between source and target nodes are

concurrent logical connections (Final Office Action, ¶8). Paragraph [0026], however, merely discloses that network routing devices are interconnect fabric modules (IFM's) capable of being dynamically configured to interconnect their respective ports. The paragraph also notes that the IFM's can communicate data using the Fibre Channel or Infiniband standards. At nowhere in this paragraph is there any discussion of establishing multiple concurrent logical connections between the same pair of nodes.

Paragraph [0033] describes Fig. 1 and similarly discloses IFM's that communicate with one another using virtual addresses. Also similar to paragraph [0026], however, paragraph [0033] includes no discussion of establishing multiple concurrent logical connections between the same pair of nodes.

Paragraph [0036] describes Fig. 3 and merely discloses a network topology discovery process that a network manager uses to determine a network topology. Each device sends a message through each of its ports to the other devices connected to such ports, and receives indications of what ports on the connected devices are connected to the device. From this information, a mapping of ports between IFM's can be created. Precisely how this paragraph is related to claim 1, however, is unclear, as there is no discussion of establishing logical connections, much less multiple concurrent logical connections between a pair of nodes.

From a review of paragraphs [0026], [0033] and [0036], as well as from a review of the Examiner's response to Applicants' arguments in ¶45 of the Final Office Action, it appears the Examiner is still under the mistaken impression that a network topology that includes an interconnected set of devices defines multiple concurrent logical connections between devices. However, as discussed above, the mapping of virtual addresses to unique connection paths in *Foster* necessitates that a single virtual address to be used to communicate between any pair of devices in the *Foster* network. Thus, to the extent that a virtual address might correspond to a logical connection, *Foster* still does not disclose the use of multiple virtual addresses to concurrently communicate over multiple paths between the same pair of nodes.

Claim 1 is therefore novel over *Foster*, as *Foster* lacks any disclosure of establishing multiple concurrent logical connections between a pair of nodes.

Applicants also respectfully submit that claim 1 is novel over *Foster* because *Foster* lacks any disclosure of a "cluster data port" within the context of Applicants' invention. As discussed at page 6, lines 2-6, a cluster data port may be considered to be:

> an abstracted transport service suitable for encapsulating and managing the establishment of multiple network connections between a source node, a target node and one or more backup nodes in such a manner that a cluster data port is effectively utilized as single data port from the perspective of a user program.

Applicants can find no disclosure in *Foster* directed to any type of "abstracted transport service" capable of managing multiple logical connections between nodes, much less a service capable of managing multiple concurrent logical connections as recited in claim 1. Accordingly, claim 1 is novel over *Foster* for this additional reason.

Applicants therefore respectfully submit that claim 1 is novel over *Foster*, and that the rejection should be withdrawn.

Applicants also respectfully submit that claim 1 is also non-obvious over *Foster* and the other prior art of record, as neither *Foster*, nor any of the other art of record, appreciates the desirability of a cluster data port that establishes multiple concurrent logical connections between a source node and a destination node. As noted above, *Foster* only discloses changing the path associated with a virtual address, while at all times maintaining a mapping between a virtual address and a single path between a source node and a destination node. No objective evidence has been provided establishing any motivation in the art to modify *Foster* to incorporate multiple concurrent logical connections between a source node and a destination node, and associated with a cluster data port, and as such, a *prima facie* case of obviousness has not been established with respect to claim 1.

With respect to independent claims 16, 31 and 35, each of these claims recites in part a cluster data configured in a similar manner to that recited in claim 1, and thus including the establishment of multiple concurrent logical connections between a source node and a target node. Accordingly, Applicants submit that claims 16, 31 and 35 are likewise novel and non-obvious over *Foster*.

Reversal of the Examiner's rejections of independent claims 1, 16, 31 and 35, as well as allowance of these claims and of claims 3-15, 18-30, 32-34 and 36 that depend therefrom, are therefore respectfully requested.

*Dependent Claims 3-4 and 18-19*

Dependent claims 3-4 and 18-19 are not argued separately.

*Dependent Claims 6, 21 and 32*

Claim 6 depends from claim 1, and additionally recites that communicating data from the source node to the target node includes performing load balancing in the cluster data port to distribute the data among the plurality of connection paths. Claims 21 and 32 respectively depend from claims 16 and 31, and similarly recite the concept of performing load balancing among the plurality of connection paths between the source and target nodes.

As such, these claims are all directed to the concept of performing load balancing between multiple connection paths coupled to the same source and target nodes.

In rejecting these claims, the Examiner relies on paragraph [0024] of *Foster*. This paragraph, however, refers to performing load balancing between destination nodes that provide the same service. The relevant portion of the paragraph states that "if multiple destination nodes provide the same functionality, then the network manager may implement node load balancing by dynamically changing a path so that data will be sent to a different destination node."

Consequently, contrary to the Examiner's assertion, *Foster* does not disclose load balancing between multiple connection paths coupled between the same nodes. In effect, *Foster* discloses load balancing among multiple destination nodes, while claims 6, 21 and 32 are directed to load balancing among multiple connection paths to the same destination node. Claims 6, 21 and 32 are therefore novel over *Foster*, and the rejections thereof should be reversed.

*Dependent Claims 8 and 23*

Dependent claims 8 and 23 are not argued separately.

*Dependent Claims 9, 24 and 33*

Claim 9 depends from claim 1, and additionally recites, with the cluster data port, switching data flow from the target node to a backup target node in response to an inability to communicate with the target node. Claims 24 and 33 respectively depend from claims 16 and 31, and similarly recite the concept of switching data flow over to a backup target node in response to an inability to communicate with the target node.

In rejecting these claims, the Examiner relies on paragraph [0035] of *Foster*. However, this paragraph merely discloses a network topology discovery process, and appears to be completely irrelevant to any functionality associated with failover between target nodes. *Foster* does disclose failing over to different paths in paragraph [0024]; however, the claims at issue are directed to failing over to different target nodes. *Foster* therefore does not disclose switching over to a different target node in response to an inability to communicate with a target node. Moreover, it is important to note that the switch over recited in these claims is performed by a cluster data port, a construct that is also not disclosed by *Foster*, as described above in claim 1. Claims 9, 24 and 33 are therefore novel over *Foster*, and the rejections thereof should be reversed.

*Dependent Claims 10, 12, 14, 25, 27, 29 and 36*

Dependent claims 10, 12, 14, 25, 27, 29 and 36 are not argued separately.

**D. Claims 5, 7, 11, 13, 20, 22, 26 and 28 are non-obvious over *Foster* and *Modi*.**

Claims 5, 7, 11, 13, 20, 22, 26 and 28 are rejected as being unpatentable over *Foster* in view of *Modi* cited by the Examiner. A *prima facie* showing of obviousness, however, requires that the Examiner establish that the differences between a claimed invention and the prior art "are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art." 35 U.S.C. §103(a). Such a showing requires that all claimed features be disclosed or suggested by the prior art. Such a showing also requires objective evidence of the suggestion, teaching or motivation to combine or modify prior art references, as "[c]ombining prior art references without evidence of such a suggestion, teaching

or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability -- the essence of hindsight." In re Dembiczak, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

Applicants respectfully submit that, in the instant case, the Examiner has failed to establish a *prima facie* case of obviousness as to claims 5, 7, 11, 13, 20, 22, 26 and 28, as such, the rejections thereof should be reversed.

*Dependent Claims 5 and 20*

Claims 5 and 20 are not separately argued.

*Dependent Claims 7 and 22*

Claims 7 and 22 respectively depend from claims 1 and 16, and each additionally recites that each logical connection comprises a TCP connection. In rejecting these claims, the Examiner admits that *Foster* does not disclose this additional feature. Instead, the Examiner relies upon *Modi*, arguing that *Modi* discloses TCP connections at col. 5, lines 40-46. It is important to note, however, that *Modi* discloses only single TCP connections between individual clients and a cluster. Thus, as with *Foster*, *Modi* does not disclose the establishment of multiple concurrent logical connections, much less multiple concurrent TCP connections, between a source node and a destination node. Claims 7 and 22 are thus non-obvious over the combination of *Foster* and *Modi*. Reversal of the Examiner's rejections of these claims are therefore respectfully requested.

*Dependent Claims 11 and 26*

Claim 11 depends from claim 9, and additionally recites notifying a client of the cluster data port service of the inability to communicate with the target node, where switching data flow from the target node to a backup node is performed in response to initiation of a cluster data port failover by the client. Claim 26 depends from claim 24 and recites similar subject matter.

In rejecting these claims, the Examiner admits that *Foster* does not disclose these additional features. Instead, the Examiner relies upon *Modi*, and in particular col. 5, lines 28-39

of the reference, arguing that *Modi* discloses switching over to a backup secondary interface node in the event of a failover.

It is important to note, however, that these claims recite a switchover that occurs in response to initiation of a cluster data port failover by a client. Applicants can find no disclosure or suggestion of any such client-initiated switchover, and as such, Applicants submit that the Examiner has failed to establish a *prima facie* case of obviousness with respect to these claims. Claims 11 and 26 are thus non-obvious over the combination of *Foster* and *Modi*. Reversal of the Examiner's rejections of these claims are therefore respectfully requested.

*Dependent Claims 13 and 28*

Claim 13 depends from claim 1, and additionally recites, with the cluster data port, resetting a logical connection between the source node and the target node in response to an inability to communicate with the target node. Claim 28 depends from claim 16 and recites similar subject matter.

In rejecting these claims, the Examiner admits that *Foster* does not disclose these additional features. Instead, the Examiner relies upon *Modi*, and in particular col. 8, lines 26-41 of the reference, arguing that *Modi* discloses resetting a logical connection. The cited passage, however, merely discloses a checkpointing process to ensure that configuration data is retained in secondary servers to enable a secondary server to take over in the event of a failure. Applicants can find no disclosure or suggestion of resetting a logical connection in the cited passage, and as such, Applicants submit that the Examiner has failed to establish a *prima facie* case of obviousness with respect to these claims. Claims 13 and 28 are thus non-obvious over the combination of *Foster* and *Modi*. Reversal of the Examiner's rejections of these claims are therefore respectfully requested.

## CONCLUSION

In conclusion, Applicants respectfully request that the Board reverse the Examiner's rejections of claims 1, 3-16 and 18-36, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at 513/241-2324.

Moreover, if any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

WOOD, HERRON & EVANS, L.L.P.

Date: November 21, 2006

By: /Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323

2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
(513) 241-2324 - Telephone
(513) 241-6234 - Facsimile

## VIII. CLAIMS APPENDIX: CLAIMS ON APPEAL (S/N 10/626,086)

1. (Once Amended) A method of communicating between a source and a target node in a clustered computer system, the method comprising:

establishing a cluster data port between the source node and a target node, the cluster data port configured to select among a plurality of connection paths between the source node and the target node, and to selectively switch over data flow from the target node to a backup target node, wherein establishing the cluster data port includes establishing multiple concurrent logical connections between the source node and the target node, each logical connection configured to communicate data over a connection path among the plurality of connection paths; and

communicating data from the source node to the target node using the cluster data port.

2. (Canceled).

3. (Once Amended) The method of claim 1, wherein establishing the cluster data port includes exchanging between the source and target nodes network addresses associated with at least one of the source and target nodes.

4. (Once Amended) The method of claim 1, wherein establishing the cluster data port includes registering a client that is resident on the source node with the cluster data port.

5. (Original) The method of claim 4, wherein registering the client comprises identifying to the data cluster port at least one callback function associated with the client, wherein the cluster data port is configured to notify the client of a data port event by calling the callback function.

6. (Original) The method of claim 1, wherein communicating data from the source node to the target node includes performing load balancing in the cluster data port to distribute the data among the plurality of connection paths.

7. (Once Amended) The method of claim 1, wherein each logical connection comprises a TCP connection.

8. (Original) The method of claim 1, wherein the cluster data port is configured to communicate data between the source and target nodes according to a messaging protocol selected from the group consisting of an asynchronous messaging protocol and a synchronous messaging protocol.

9. (Original) The method of claim 1, further comprising, with the cluster data port, switching data flow from the target node to a backup target node in response to an inability to communicate with the target node.

10. (Original) The method of claim 9, wherein switching data flow from the target node to the backup target node includes establishing a logical connection between the source node and the backup target node.

11. (Original) The method of claim 9, further comprising notifying a client of the cluster data port service of the inability to communicate with the target node, wherein switching data flow from the target node to a backup node is performed in response to initiation of a cluster data port failover by the client.

12. (Original) The method of claim 9, wherein switching data flow from the target node to a backup node is initiated by the cluster data port.

13. (Original) The method of claim 1, further comprising, with the cluster data port, resetting a logical connection between the source node and the target node in response to an inability to communicate with the target node.

14. (Original) The method of claim 1, further comprising communicating data from the target node to the source node using the cluster data port.

15. (Original) The method of claim 1, wherein the target node is remote from the source node, and wherein communicating data from the source node to the target node includes communicating mirror data to support remote mirroring between the source and target nodes.

16. (Once Amended) An apparatus, comprising:

a memory;

at least one processor; and

program code resident in the memory and configured for execution on the at least one processor to implement a cluster data port for a clustered computer system of the type including a plurality of nodes, the cluster data port configured to support communication between a source node and a target node among the plurality of nodes, the cluster data port further configured to select among a plurality of connection paths between the source node and the target node, and to selectively switch over data flow from the target node to a backup target node, wherein the cluster data port is configured to establish multiple concurrent logical connections between the source node and the target node, each logical connection configured to communicate data over a connection path among the plurality of connection paths.

17. (Canceled).

18. (Once Amended) The apparatus of claim 16, wherein the cluster data port is configured to exchange between the source and target nodes network addresses associated with at least one of the source and target nodes.

19. (Once Amended) The apparatus of claim 16, wherein the cluster data port is configured to register a client that is resident on the source node.

20. (Original) The apparatus of claim 19, wherein the cluster data port is configured to receive from the client at least one callback function associated with the client during registration

of the client, wherein the cluster data port is configured to notify the client of a data port event by calling the callback function.

21. (Original) The apparatus of claim 16, wherein the cluster data port is configured to load balance data communicated between the source and target nodes to distribute the data among the plurality of connection paths.

22. (Once Amended) The apparatus of claim 16, wherein each logical connection comprises a TCP connection.

23. (Original) The apparatus of claim 16, wherein the cluster data port is configured to communicate data between the source and target nodes according to a messaging protocol selected from the group consisting of an asynchronous messaging protocol and a synchronous messaging protocol.

24. (Original) The apparatus of claim 16, wherein the cluster data port is configured to switch data flow from the target node to a backup target node in response to an inability to communicate with the target node.

25. (Original) The apparatus of claim 24, wherein the cluster data port is configured to switch data flow from the target node to the backup target node by establishing a logical connection between the source node and the backup target node.

26. (Original) The apparatus of claim 24, wherein the cluster data port is further configured to notify a client of the cluster data port of the inability to communicate with the target node, and wherein the cluster data port is configured to switch data flow from the target node to a backup node in response to initiation of a cluster data port failover by the client.

27. (Original) The apparatus of claim 24, wherein the cluster data port is configured to initiate the switch of data flow from the target node to a backup node.

28. (Original) The apparatus of claim 16, wherein the cluster data port is configured to reset a logical connection between the source node and the target node in response to an inability to communicate with the target node.

29. (Original) The apparatus of claim 16, wherein the cluster data port is configured to support bidirectional communication between the source and target nodes.

30. (Original) The apparatus of claim 16, wherein the target node is remote from the source node, and wherein the cluster data port is configured to communicate mirror data from the source node to the target node to support remote mirroring between the source and target nodes.

31. (Once Amended) A clustered computer system, comprising:

a plurality of nodes; and

a cluster data port resident on at least one of the plurality of nodes and configured to support communication between a source node and a target node among the plurality of nodes, the cluster data port configured to select among a plurality of connection paths between the source node and the target node, and to selectively switch over data flow from the target node to a backup target node, wherein the cluster data port is configured to establish multiple concurrent logical connections between the source node and the target node, each logical connection configured to communicate data over a connection path among the plurality of connection paths.

32. (Original) The clustered computer system of claim 31, wherein the cluster data port is configured to select among the plurality of connection paths using a load balancing algorithm.

33. (Original) The clustered computer system of claim 31, wherein the cluster data port is configured to switch over data flow from the target node to the backup target node in response to an inability of the source node to communicate with the target node.

34. (Original) The clustered computer system of claim 31, wherein the target node is remote from the source node, and wherein the cluster data port is configured to communicate mirror data from the source node to the target node to support remote mirroring between the source and target nodes.

35. (Once Amended) A program product, comprising:

program code configured to implement a cluster data port for a clustered computer system of the type including a plurality of nodes, the cluster data port configured to support communication between a source node and a target node among the plurality of nodes, the cluster data port further configured to select among a plurality of connection paths between the source node and the target node, and to selectively switch over data flow from the target node to a backup target node, wherein the cluster data port is configured to establish multiple concurrent logical connections between the source node and the target node, each logical connection configured to communicate data over a connection path among the plurality of connection paths; and

a signal bearing medium bearing the program code.

36. (Original) The program product of claim 35, wherein the signal bearing medium includes at least one of a recordable and a transmission medium.

## IX. EVIDENCE APPENDIX



None.

## X. RELATED PROCEEDINGS APPENDIX

10/626,086

None.